Image Caption Generator Report

Wardah Ijaz Damon Ma Modaser Mojadiddi Ziad Zananiri
Department of Mathematics, Computer Science and Statistics (MCS)
University of Toronto - Mississauga

Abstract

Translating visual content into natural language remains challenging due to the high dimensionality of images and the ambiguity of linguistic descriptions. While modern vision–language models achieve strong captioning performance, encoder–decoder frameworks remain valuable for analyzing how visual representations interact with sequence models. This project investigates whether CLIP's pre-trained vision encoder provides a stronger visual representation than conventional CNN encoders such as ResNet when paired with a Transformer-based captioning decoder. CLIP offers semantically rich, text-aligned embeddings that may yield advantages over features learned solely from visual supervision. To test this, we construct an image captioning model that substitutes the standard CNN encoder with CLIP while keeping the Transformer decoder fixed, enabling a controlled comparison against a ResNet–Transformer baseline. Using the Flickr8k dataset, we evaluate caption quality with BLEU-4, METEOR, CIDEr, and BERTScore. Our goal is to determine whether CLIP's multimodal representations measurably improve caption accuracy and descriptiveness within a modern attention-based pipeline.

1 Introduction

Vision is the primary means through which humans perceive and interpret the world [18]. A key aspect of this ability is the capacity to translate visual input into language to communicate effectively. Replicating this ability in artificial intelligence systems remains challenging because visual scenes are high-dimensional, context-dependent, and often ambiguous. Image captioning therefore provides an important testbed for studying grounded language generation.

The standard approach in industry settings follows an encoder-decoder framework due to its simplicity and strong performance [4]. This uses a Convulational Neural Network (CNN) such as ResNet to encode the visual information and a Recurrent Neural Network (RNN), often a Long Short-Term Memory (LSTM) decoder, to generate captions sequentially. While effective, this pipeline has two key limitations: CNN encoders learn purely visual features that may lack semantic alignment with language, and LSTMs struggle to model long-range dependencies and global context.

Recent advances in vision–language modeling have introduced models such as CLIP, which learn to align images and text within a shared embedding space [6]. This alignment allows CLIP to produce visual representations that carry richer semantic information than those obtained from CNNs trained solely on visual classification tasks [5]. Likewise, Transformer-based decoders address the limitations of LSTMs by capturing long-range dependencies and contextual relationships through self-attention rather than sequential recurrence.

Building on these developments, this project explores a modern alternative to the traditional CNN–LSTM pipeline by pairing CLIP's text-aligned visual features with a Transformer captioning decoder. By comparing this CLIP–Transformer architecture directly against a ResNet–LSTM baseline within the same encoder–decoder framework, we aim to assess how much CLIP's multimodal representations contribute to improvements in caption quality, descriptiveness, and semantic accuracy.

2 Background and Related Work

2.1 Image Captioning with CNN–LSTM Models

Traditionally, image captioning approaches use a Convoluational Neural Network (CNN) to encode the input into feature embeddings followed by a Recurrent Neural Network (RNN), generally Long Short-Term Memory (LSTM), to generate captions sequentially[11].

Although the CNN-LSTM pipeline has historically served as a strong baseline, there are some limitations associated with it [9]:

- They rely on a single visual embedding and cannot focus on specific image regions, often resulting in less relevant words.
- LSTMs work sequentially which prevents parallel processing, resulting in slower training and inference times.
- Maintaining recurrent hidden states increases memory usage, making longer captions and large datasets harder to scale.

2.2 Visual-Language Models and Transformers

Transformer architectures offer a promising alternative. Instead of treating visual and textual modalities separately, these models jointly learn representations from both images and texts. Vision-language transformers, such as the Vision Transformer (OpenAI, 2021) for images and text transformers like GPT for language, can be integrated into frameworks that process and align multimodal input together [8].

For example, models like CLIP (Contrastive Language-Image Pre-training) learn to associate images and textual descriptions by training on large-scale datasets of image-text pairs using contrastive learning objectives [8]. This approach enables zero-shot learning across various tasks, including image captioning.

Transformer based models address a lot of the limitations faced with the CNN-LSTM pipeline:

- Transformers use a cross-attention mechanism which allows the decoder to access information from the encoder.
- It allows the use of the self-attention mechanism which allows the data to be split into multiple batches and processed concurrently. [15].
- Transformers efficiently handle longer sequences without maintaining hidden states over time.

2.3 Prior Work

A notable study regarding Transformer-based image captioning is **Meshed Memory Transformer ($\mathbf{M}^2$ Transformer)** proposed by Cornia et al, 2020 [2]. It introduces two concepts: memory-augmented encoding and meshed decoding. The encoder utilizes memory-augmented attention, which uses learnable memory vectors to capture prior semantic relationships between image regions. The decoder introduces a meshed connectivity pattern, allowing each decoding layer to consider outputs from multiple encoder layers rather than only the final one. This design enables the model to combine low-level and high-level visual information when generating captions leading to great performance on the MS-COCO dataset. Our work builds on this idea of deep multimodal interaction but replaces region-based visual features with CLIP's pretrained vision-language embeddings, enabling more general visual understanding without relying on object detectors.

Another interesting advancement in the Image Caption Generation space is the **Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation (BLIP)** framework [15]. This work emphasizes that vision transformers, such as CLIP, excel at understanding and aligning visual and textual representations but are not able to generate fluent natural language descriptions. BLIP addresses this limitation through the Multimodal Encoder-Decoder (MED) architecture. The encoder operates similarly to CLIP for learning image–text alignments, while the decoder handles text generation. The model is trained using three objectives: Image–Text

Contrastive (ITC) for alignment, Image–Text Matching (ITM) for pairwise relevance, and Language Modeling (LM) for caption generation. In addition to the objectives, it also adopts a data bootstrapping strategy to refine noisy image–text pairs. BLIP notably achieves strong performance on captioning benchmarks. However, it also relies on a large curated datasets and specialized training objectives. Our project explores a simpler CLIP-based captioning pipeline that directly uses CLIP's pretrained embeddings as visual features in a lightweight decoder. This approach investigates whether strong captioning performance can be achieved without task-specific multimodal pre-training or complex objective combinations, and how CLIP's vision–language space can be repurposed for generation.

2.4 Summary of Positioning

Overall, prior work shows notable progression from CNN-LSTM pipelines to region-based transformer encoders. More recently, there has been development regarding large scale multimodal pre-training frameworks like BLIP. Our method aims to extend this trajectory by examining whether CLIP's pretrained and general-purpose embeddings can offer a simpler alternative to more computational and resource heavy pipelines while maintaining strong results.

3 Data Description and Preprocessing Steps

We use the Flickr8k dataset [10], available on Kaggle, which consists of 8,000 natural images, each annotated with five human-generated captions describing salient objects, scenes, and actions. Although relatively small, this dataset is sufficient for our study because we leverage a pretrained vision-language model (CLIP) to extract semantically rich image embeddings. We plan to split the dataset into 6,000 training, 1,000 validation, and 1,000 test images. Table 1 summarizes the dataset statistics.

Table 1: Flickr8k dataset statistics and data splits

Subset	Images	Captions	Notes
Training	6,000	30,000	5 captions per image
Validation	1,000	5,000	5 captions per image
Test	1,000	5,000	5 captions per image

3.1 Image Preprocessing

All images are resized to 224×224 pixels and normalized using predefined mean and standard deviation values to ensure compatibility with the visual encoder described in Section 4. These transformations standardize input images and prepare them for efficient feature extraction.

3.2 Text Preprocessing

Text preprocessing is generally similar to our coursework.

- Convert all captions to lowercase and remove punctuation.
- Tokenize captions and build `word2idx` and `idx2word` mappings.
- Restrict the vocabulary to words appearing at least five times in the training captions to reduce noise.
- Pad or truncate sequences to a fixed length for batch processing.

This preprocessing ensures consistency and reproducibility, allowing the model to focus on learning the mapping between image embeddings and natural language captions.

3.3 Notes on Dataset Size and Pretrained Models

Although Flickr8k is moderate in size, the use of **pretrained CLIP embeddings** mitigates data scarcity issues. If computational resources allow, we plan to extend experiments to the Flickr30k

dataset to test scalability and generalization. We obtained educational access to Flickr30k from University of Illinois.

4 Model Architecture

The architecture mainly consists of CLIP image encoder, an image projection layer, and a Transformer-based caption decoder.

1. CLIP Image Encoder:
   Instead of using a traditional CNN to extract features, we employ CLIP's pretrained image encoder (ViT-B/32) [1]. During training, CLIP can accept both images and text to learn a joint embedding space, where semantically similar images and text are grouped closely. However, text input is not required at test time. For each image, the built-in ResNet extracts low-level visual features, such as edges, objects, and textures, which are then projected into a 512-dimensional vector in the shared embedding space to produce a rich semantic representation of the image [8]. Preprocessing steps are described in Section 3.1.
2. Image Projection Layer:
   To match the Transformer decoder's hidden dimension ($d_{\text{model}} = 512$), the CLIP embedding is passed through a linear projection layer. The projected vector serves as the memory input for the Transformer decoder, allowing it to attend to the image's semantic content while generating captions. Formally, for a CLIP embedding $x_{\text{CLIP}} \in \mathbb{R}^{512}$, the memory vector is computed as
   $$\text{memory} = W_{\text{proj}} x_{\text{CLIP}} + b_{\text{proj}} \in \mathbb{R}^{d_{\text{model}}}.$$
3. Transformer Decoder:
   The Transformer decoder is responsible for generating captions one token at a time, conditioned on the projected CLIP image embedding. The decoder consists of stacked Transformer layers that combine self-attention, cross-attention to image features, and feedforward networks. The input caption tokens are mapped to dense vector representations via a learned embedding layer, after which sinusoidal positional encodings are added to encode sequential order information. [13].

   During training, a causal (look-ahead) mask prevents tokens from attending to future positions. The decoder uses cross-attention over the projected image features to condition each predicted word on the image content.

   The final outputs of the decoder are passed through a linear layer and softmax function to produce a probability distribution over the vocabulary for the next word at each time step. The model is trained using teacher forcing, where ground-truth tokens are provided as inputs during training.
4. Hyperparameters
   - **batch_size:** 32
   - **num_epochs:** 10
   - **learning_rate:** 1e-4
   - **d_model:** 512
   - **nhead:** 8
   - **num_layers:** 6
   - **dim_feedforward:** 2048
   - **dropout:** 0.1
   - **max_len:** 50
   - **vocab_size:** 5000
   - **min_freq:** 2
   - **train_split:** 0.8

   Several hyperparameters were experimented with to find the final configuration we decided on based on results seen. Below is key hyperparameters we experimented with. For other hyperparameters like attention heads and feedforward dimension we used the commonly adoped values which are motivated by the architecture and previous works.

Vocabulary Size and Minimum Frequency. We limit the vocabulary to the 5,000 most frequent tokens in the training captions and discard words that occur fewer than two times. 5000 most common words allows us to capture 90%+ of all words in the dataset and at the same time we avoid rare tokens that do not occur enough for the model to learn them.

Hidden Dimension. The choice of $d_{\text{model}} = 512$ is motivated by two considerations: it matches the dimensionality of CLIP embeddings, simplifying the projection step, and it provides sufficient model capacity for capturing semantic relationships in captions while remaining computationally efficient compared to 768.

Transformer Depth. We use a 6 layer Transformer decoder. This depth level has a good ratio between learning and training time. Less layers can underfit and struggle with long-range sentence structures, while more layers have diminishing returns on our dataset and significantly increase training time.

Learning Rate. We train using Adam with a learning rate of 1×10^{-4},. Lower rates slowed convergence significantly and higher learning rates caused unstable loss spikes.

5. Inference (Caption Generation)
   During inference, the trained model generates captions in an autoregressive manner, producing one word at a time given an input image. First, the image is passed through the CLIP image encoder and the projection layer to obtain a fixed-dimensional memory vector. Caption generation is initialized with a special <START> token.

   At each decoding step, the Transformer decoder receives the previously generated tokens along with the image memory and predicts a probability distribution over the vocabulary for the next token. The word with the highest probability (greedy decoding) is selected and appended to the caption. This process continues until an <END> token is produced or a maximum caption length is reached.

Each component of the proposed model architecture can be visualized by the following diagram:

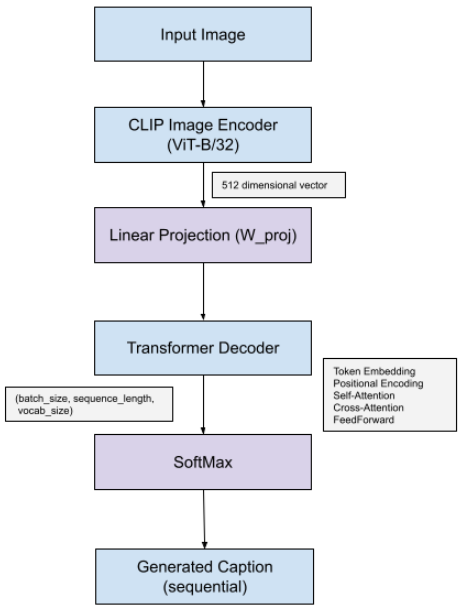

Figure 1: Model Architecture

5 Quality of Results

5.1 Performance Benchmarks Used

To compare the quality of results for our machine model, we will use the following metrics:

- **BLEU-4:** A well-known benchmark used to judge translation and captioning performance. It is able to measure n-gram overlap bewteen a generated caption and its reference captions [3].
- **METEOR:** Also well-known, this score is used to determine how close the generated caption is to the its reference caption. This metric can compare word orders and only requires the words to be share the same root form or that the words are synonyms to each other. This complements the BLEU-4 score as it will not penalize the use of very similar words [12].
- **CIDEr:** This metric measures the similarity of a generated caption to a consensus of human reference captions. It computes the average cosine similarity between the Term Frequency-Inverse Document Frequency (TF-IDF) weighted n-grams of the candidate caption and the reference set, capturing how well the caption aligns with the majority of human descriptions [5].
- **BERTScore:** This metric is used to determine how much the generated caption's overall meaning differs from the reference captions [17].

5.2 Models Compared

To understand how the quality of our model's predicted captions are, we use the above mentioned benchmarks and the following pre-trained models:

- **Show and Tell (NIC):** This was one of the first image captioning models to be produced, which will allow us to determine the improvements of using CLIP and a Transformer over older CNN and LSTM models. This model was trained using the MS-COCO data set [14].
- **Show, Attend and Tell (Hard Attention):** This model was the first image captioning model to introduce the use of attention. By comparing this to our model, we hope to demonstrate improvements in captioning accuracy over CNN encoders and LSTM decoders with attention [16]. For consistency in comparison we will be benchmarking this model when it was trained on the Flicker8k dataset.
- **ClipCap (CLIP + GPT2 Transformer):** This model represents a modern approach that, like our work, utilizes CLIP embeddings for visual encoding. However, instead of training a decoder from scratch, it employs a lightweight mapping network to translate visual features into a prefix for a frozen GPT-2 language model. By comparing our model to the ClipCap Transformer variant, we can evaluate the effectiveness of our lightweight decoder architecture against a method that adapts a massive, pre-trained Large Language Model (LLM) for the same task [20].

Important Note: We are aware that some of the models are using for comparison are trained on different data sets, and may not be a perfect point of comparison. We have chosen to compare with these models because of their historical significance in image captioning models, and there are very few well-known models trained with the Flickr8 dataset. Additionally, these three models use the MS-COCO data set which is much larger than Flickr8k. We aim to demonstrate similar results with a much smaller data set.

5.3 Benchmarking Scores

Metric	Our Model (CLIP + Transformer)	Show and Tell (NIC)	Show, Attend and Tell (Hard Attention)	ClipCap (CLIP + GPT2)
BLEU-4 (NLTK)	22.28	27.7	21.3	33.53
METEOR	28.64	23.7	20.30	27.45
CIDER	62.28	85.5	–	113.08
BERTScore	63.03	–	–	43.05*

Note: Some of the models do not have recorded values for some of the benchmarking scores we are using for comparison (Primarily for BERTScore). This is because these models did not exist before the introduction of these benchmarks. Missing score values are marked by ("–").

6 Discussion and Analysis

6.1 Analysis of Benchmarking Results

Our evaluation demonstrates that replacing the traditional CNN encoder with CLIP's pretrained embeddings yields significant semantic improvements, even when trained on the smaller Flickr8k dataset.

Comparison against Traditional Baselines: Our model achieves a METEOR score of 28.64, outperforming both *Show and Tell* (23.7) and *Show, Attend and Tell* (20.30). Since METEOR correlates well with human judgment regarding synonymy and semantic relevance, this suggests that the CLIP embeddings successfully capture high-level semantic concepts better than ResNet-based encoders. While our BLEU-4 score (22.28) is slightly lower than *Show and Tell* (27.7), it surpasses *Show, Attend and Tell* (21.3). This trade-off, lower n-gram precision but higher semantic recall, is expected when leveraging CLIP, as the encoder prioritizes semantic alignment over exact visual structural replication.

Comparison against Modern Baselines (ClipCap): A direct comparison with ClipCap reveals important nuances regarding architecture and dataset scale.

While ClipCap achieves a significantly higher BLEU-4 score (33.53) compared to our model (22.28), our model surpasses ClipCap in METEOR (28.64 vs. 27.45). This divergence is instructive: BLEU prioritizes exact n-gram overlap, an area where ClipCap excels because it leverages a massive, pre-trained GPT-2 decoder that is highly optimized for linguistic fluency and probability. However, METEOR accounts for synonyms and semantic paraphrasing. Our higher METEOR score suggests that our model, which trains a decoder from scratch over CLIP embeddings, may be more effective at capturing the raw semantic "gist" of the image and using valid synonyms, even if it lacks the rigid n-gram fluency of a pre-trained Large Language Model.

Regarding CIDEr, the reported score for ClipCap on the massive MS-COCO dataset is 113.08. However, when the ClipCap authors tested their model on the *nocaps* dataset (evaluating generalization to unseen objects), they reported a CIDEr score of 65.83. Our model achieves a CIDEr score of 62.28 on Flickr8k. This result is highly competitive with ClipCap's performance on challenging, unseen data, suggesting that our lightweight decoder architecture creates strong caption consensus relative to the dataset size.

BERTScore Analysis: We report a BERTScore of 63.03. For the ClipCap baseline, the score of 43.05 (marked with *) was not available in the original paper; we calculated this value manually by running inference on a pre-trained instance of ClipCap using our validation set. The discrepancy suggests that while ClipCap generates coherent text (high CIDEr/BLEU on COCO), our model generates embeddings that may be closer in the BERT vector space to the Flickr8k references, or that the domain shift affects the pre-trained ClipCap model when evaluated strictly on Flickr8k data.

6.2 Model Limitations

Despite the strong semantic performance, our approach faces several limitations inherent to the architecture and data constraints:

- **Dataset Size and Vocabulary:** Our model was trained on Flickr8k (8,000 images), whereas baselines like Show and Tell and ClipCap utilized MS-COCO (100,000+ images). This limited training data restricts the model's vocabulary. Consequently, the model struggles with naming specific entities (e.g., celebrities, specific landmarks) or rare objects that appear infrequently in the training split. It tends to default to generic descriptions (e.g., "a person" instead of "a baseball player") when facing ambiguous inputs.
- **Spatial and Numeric Precision:** While CLIP captures the "gist" of a scene effectively, it is known to struggle with fine-grained spatial relationships and counting. Our model occasionally hallucinates the number of objects (e.g., captioning "three dogs" when there

are two) or misinterprets complex spatial prepositional phrases, a known side-effect of using a pooled semantic embedding rather than region-based features used in attention models.

- **Greedy Decoding Issues:** Our inference pipeline currently utilizes greedy decoding. While efficient, this approach can sometimes lead to repetitive loops or abrupt endings in generated captions. Implementing beam search could potentially improve fluency and reduce grammatical fragmentation.
- **Domain Specificity:** The model performs well on natural scenes similar to Flickr data but degrades on out-of-distribution images, such as diagrams, screenshots, or highly stylized art, which lack semantic anchors in the CLIP encoder's primary training distribution.

7 Ethical Considerations

Our machine learning model is trained on publicly available datasets, Flickr8k, which includes images of people and everyday scenes. Although these images are publicly available, they may contain images of identifiable individuals, and generated captions may unintentionally reveal personal information, posing potential privacy risks. Captions may also reinforce demographic or cultural stereotypes by associating certain professions or activities with specific genders or ethnic groups. Additionally, the datasets were primarily collected in the early 2010s, introducing potential temporal bias in terms of clothing, technology, and social behaviors. Many publicly available images capture staged or highly shareable moments, which could lead the model to generate captions that are overly positive, superficial, or unrepresentative of real-world diversity.

To mitigate these ethical concerns, our work is restricted to research purposes. All generated captions are explicitly marked as machine-generated. We supplement evaluation with a pre-trained Hugging Face multi-label toxicity classifier, which estimates probabilities across categories such as explicit, violent, hateful, or illegal content and measures uncertainty. This system is used only as a diagnostic tool and does not replace human judgment. Finally, we employ neutral evaluation metrics and acknowledge dataset biases. These measures are designed to ensure that our work is responsible, reproducible, and sensitive to potential harms while advancing research in image captioning.

References

[1] Aneja, Jyoti, et al. "Convolutional Image Captioning." arXiv.org, arXiv, 24 Nov. 2017, https://arxiv.org/abs/1711.09151.

[2] Cornia, Marcella, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. "Meshed-Memory Transformer for Image Captioning." arXiv, preprint, 17 Dec. 2019, https://arxiv.org/abs/1912.08226.

[3] evaluate-metric. "BLEU." Hugging Face, 2025, https://huggingface.co/spaces/evaluate-metric/bleu.

[4] Ghosh, Ankan. "Image Captioning using ResNet and LSTM." LearnOpenCV, 31 December 2024, https://learnopencv.com/image-captioning/?utm_source=chatgpt.com#aioseo-but-why-are-we-using-resnet-and-lstm-today.

[5] Vedantam, Ramakrishna, C. Lawrence Zitnick, and Devi Parikh. "CIDEr: Consensus-based Image Description Evaluation." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4566–4575.

[6] Hessel, Jack, et al. "CLIPScore: A Reference-free Evaluation Metric for Image Captioning." arXiv, 18 Apr. 2021, https://arxiv.org/abs/2104.08718.

[7] Huang, Weiquan, et al. "LLM2CLIP: Powerful Language Model Unlocks Richer Visual Representation." arXiv, version v2, Nov. 2024, https://arxiv.org/html/2411.04997v2.

[8] Li, Junnan, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation." arXiv, preprint, 28 Jan. 2022, https://arxiv.org/abs/2201.12086.

[9] OpenAI. "CLIP: Connecting Text and Images." OpenAI, 5 Jan. 2021, https://openai.com/index/clip/.

[10] Palucha, Szymon. "Understanding OpenAI's CLIP Model." Medium, 24 Feb. 2024, https://medium.com/@paluchasz/understanding-openais-clip-model-6b52bade3fa3.

[11] Quadeer, Muhammad. "Flickr8k Image Captioning using CNNs + LSTMs." Kaggle Notebooks, Kaggle, 2025, https://www.kaggle.com/code/quadeer15sh/flickr8k-image-captioning-using-cnns-lstms/notebook.

[12] Singh, Hrishikesh, Aarti Sharma, and Millie Pant. "Pixels to Prose: Understanding the Art of Image Captioning." arXiv, 28 Aug. 2024, https://arxiv.org/html/2408.15714v1#S4.

[13] Wikipedia. "METEOR." Wikipedia, The Free Encyclopedia, Wikimedia Foundation, 30 June 2024, https://en.wikipedia.org/wiki/METEOR.

[14] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention Is All You Need." arXiv, 12 June 2017, https://arxiv.org/abs/1706.03762.

[15] Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. "Show and Tell: A Neural Image Caption Generator." arXiv, 20 Apr. 2015. https://arxiv.org/pdf/1411.4555.

[16] Winland, Vanna. "What Is Self-Attention?" IBM Think, IBM, n.d., https://www.ibm.com/think/topics/self-attention.

[17] Xu, Kelvin, et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." arXiv.Org, arXiv, 19 Apr. 2016, arxiv.org/abs/1502.03044. Accessed 10 Dec. 2025.

[18] Zhang, Tianyi, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. "BERTScore: Evaluating Text Generation with BERT." arXiv, 21 Apr. 2019, arxiv.org/abs/1904.09675.

[19] ZEISS Vision Care. "Why good vision is so important." 16 October 2021, https://www.zeiss.ca/vision-care/en/eye-health-and-care/health-prevention/why-good-vision-is-so-important.html.

[20] Mokady, Ron, Amir Hertz, and Amit H. Bermano. "ClipCap: CLIP Prefix for Image Captioning.", 2021. https://arxiv.org/pdf/2111.09734